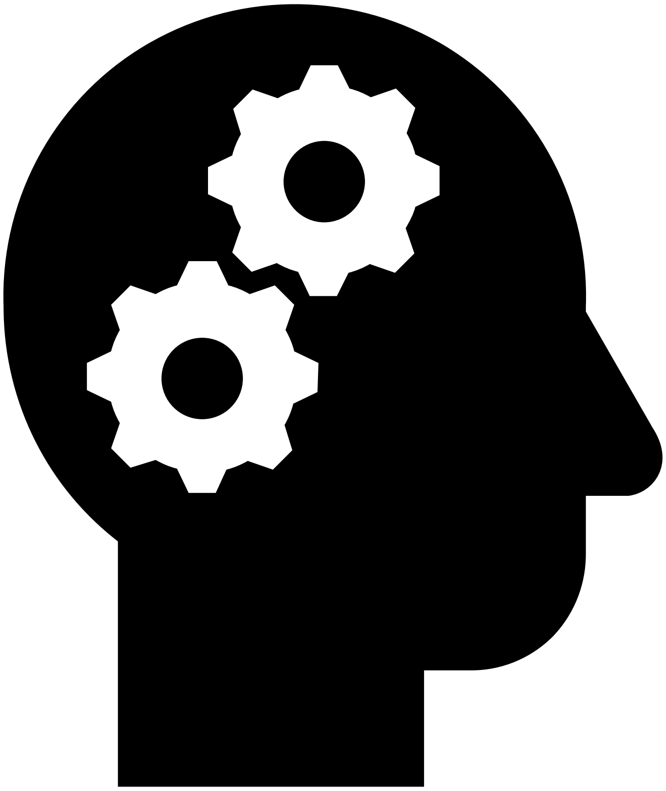


DGA Machine Learning

SPLUNK'S DGA MACHINE LEARNING APPLICATION

Andey Ng | Unum Cybersecurity Intern | 2019

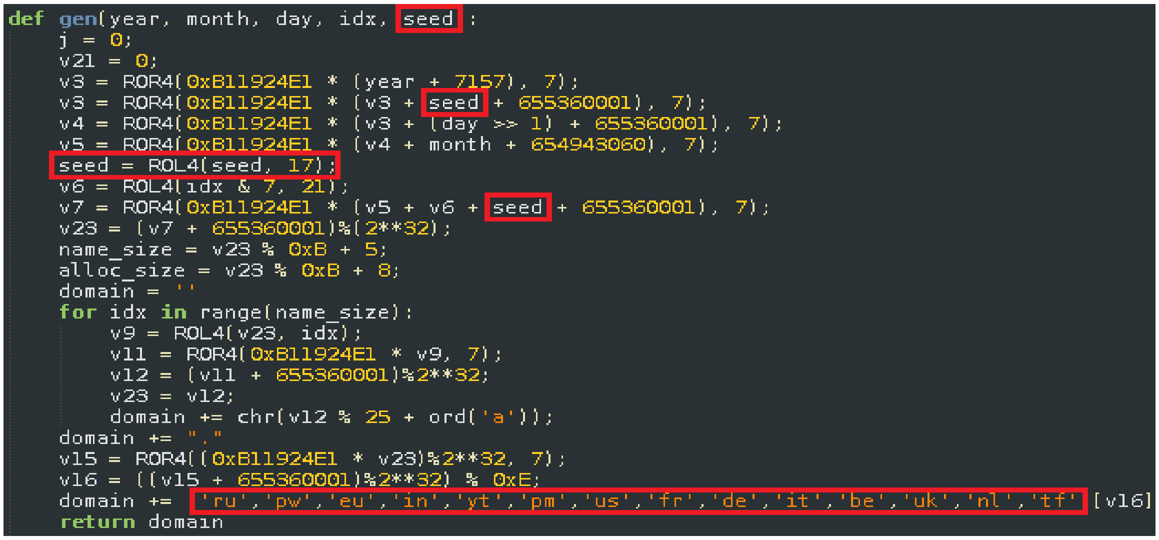
Abstract

This paper explains a high-level understanding of what DGA algorithm are and why they are relevant and urgent to understand in today's cybersecurity field, and a step-by-step process of how to defend a network from DGAs using Splunk's Machine Learning Application.

I developed this application and paper through my cybersecurity internship this summer at Unum Insurance Company to help my peers understand the basics of network security and to explain what I have learned this summer.

This project touches on the intersection of software development and engineering, machine learning, computer networks, and cybersecurity tactics and has piqued my interest in how all these fields work together.

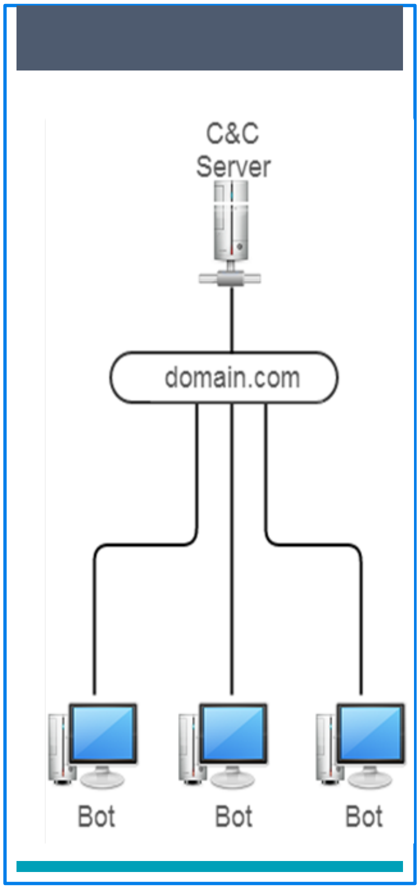
DGA algorithms are Domain Generation Algorithms that generates lists of domains that communicate with the Command and Control (C&C) Servers to allow the adversary to update and receive information.



```
def gen(year, month, day, idx, seed :
    j = 0;
    v21 = 0;
    v3 = ROR4(0xB11924E1 * (year + 7157), 7);
    v3 = ROR4(0xB11924E1 * (v3 + seed + 655360001), 7);
    v4 = ROR4(0xB11924E1 * (v3 + (day >> 1) + 655360001), 7);
    v5 = ROR4(0xB11924E1 * (v4 + month + 654943060), 7);
    seed = ROL4(seed, 17);
    v6 = ROL4(idx & 7, 21);
    v7 = ROR4(0xB11924E1 * (v5 + v6 + seed + 655360001), 7);
    v23 = (v7 + 655360001)%(2**32);
    name_size = v23 % 0xB + 5;
    alloc_size = v23 % 0xB + 8;
    domain = ''
    for idx in range(name_size):
        v9 = ROL4(v23, idx);
        v11 = ROR4(0xB11924E1 * v9, 7);
        v12 = (v11 + 655360001)%2**32;
        v23 = v12;
        domain += chr(v12 % 25 + ord('a'));
    domain += "."
    v15 = ROR4((0xB11924E1 * v23)%2**32, 7);
    v16 = ((v15 + 655360001)%2**32) % 0xE;
    domain +=  'ru','pw','eu','in','yt','pm','us','fr','de','it','be','uk','nl','tf' [v16]
    return domain
```

Because DGA's are generative and are not a brute-force list of domains, defenders cannot simply create a blacklist of domains. Thus, machine learning is a preventative tool to detect the patterns of the domains (i.e. how ratio vowels or meaning) to identify DGA domains and to automate the blacklisting of those domains.

Background



BOTNETS

Botnets are networks of internet-connected devices that attach themselves with malicious code onto domains take down servers, perform DDoS attacks, steal data, spam the network, or attain access to a victim's endpoint. Botnets perform data exfiltration, code execution, and interfering with a device's operation by attacking their servers.

Peer to Peer botnets (P2P) decentralize their network by having every bot connect and communicate with each other to remove the need for a centralized server. This makes it difficult for the defender to take down the botmaster, as there is no central location it can pinpoint.

Likewise, DGA algorithms use a similar approach where they distribute their attack on thousands of different domains, making it difficult for defenders to identify the legitimacy of a domain and to determine whether that domain contains malicious code.

Botnets are difficult to catch, as it can redirect and jump onto another non-blacklisted domain if it is to be blacklisted. In DGA attacks, botnets are used to generate thousands of pseudo-random domains.

DGA ALGORITHMS

What's a DGA?

DGA algorithms generate and configuring thousands of illegitimate domains by creating new top-level domains (TLD such as *.com* or *.net*) and domain names which contain self-updating malware processes and executable commands (which dictate the intervals that a new DGA domain should be generated). [1]

[1] More DGA Info: https://resources.infosecinstitute.com/domain-generation-algorithm-dga/

How does it Works?

In order to communicate with the botmaster[2], DGAs produce a list of candidate C&C domains. The bot then attempts to resolve these domain names by sending DNS queries until one of the domains resolves to the IP address of a C&C server. Eventually, one of the domains will receive the C&C server's IP address.

The code of DGA algorithms are smart. They often don't allow the new domain to exceed 32 characters, as it will appear to be spoofed[3]. Creating DGA algorithms requires very little work, as it requires very little variation such as *k.gov, ka.gov,* or *kaf.gov*; however, they cause a lot of work to detect and block.

Notably, the most successful and notorious DGAs are Conficker, Bobax, and Kraken. Common DGA use algorithms such as locky, chinad, or newgoz.[4]

Why are DGAs a Concern?

The objective of DGA algorithms is to evasively get one of the thousands of domains receive updates and commands from the C&C server. The malicious DGA domain contains an executable script/file attached to that navigates to the C&C server.

For example, "*ka.gov/main.py"* would execute a command from the "main.py" file where a malicious script would typically be sent to the server to communicate with it.

From there, the DGA domain can install trojans, fake anti-virus software, disable anti-virus software, encrypt data from the C&C server without the victim's knowledge.

DGA algorithms often perform fast flux techniques that register multiple IP addresses under a single domain. This evasion technique makes DGA domain change quickly and very difficult to track.

[2] Botmaster: the attacker who is controlling and orchestrating the malicious bots

[3] Spoofing: the act of disguising a communication from an unknown source as being from a known, trusted source

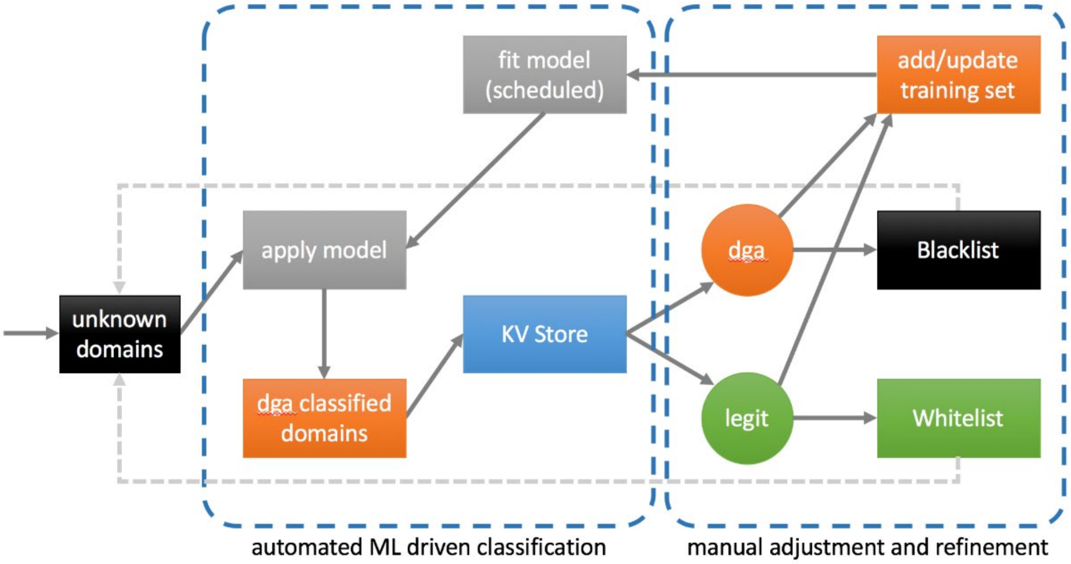
[4] Python DGA Code: https://github.com/baderj/domain_generation_algorithms

Preventative Measures with Machine Learning

Because these domains are generated very frequently (daily or hourly) and are not a brute-force list of domains, defenders cannot simply create a blacklist of domains. The defender must monitor DNS requests and responses to determine whether the domain is malicious.

The most common prevention tactic is reverse engineering these DGA algorithms with machine learning to decrypt the executable algorithm.

Thus, machine learning is used to detect the patterns of the domains (i.e. how ratio vowels or meaning) to identify DGA domains and to automate the blacklisting of those domains.



Splunk’s Machine Learning DGA Application

Splunk’s DGA Machine Learning Application uses supervised learning models that that determine the type of hack is coming in, the different types of attacks that are penetrating the system, and whether the domain is legitimate or fake.

It is a highly recommendable system, as it has built-in functions that allow analysts to give and receive feedback to the model.

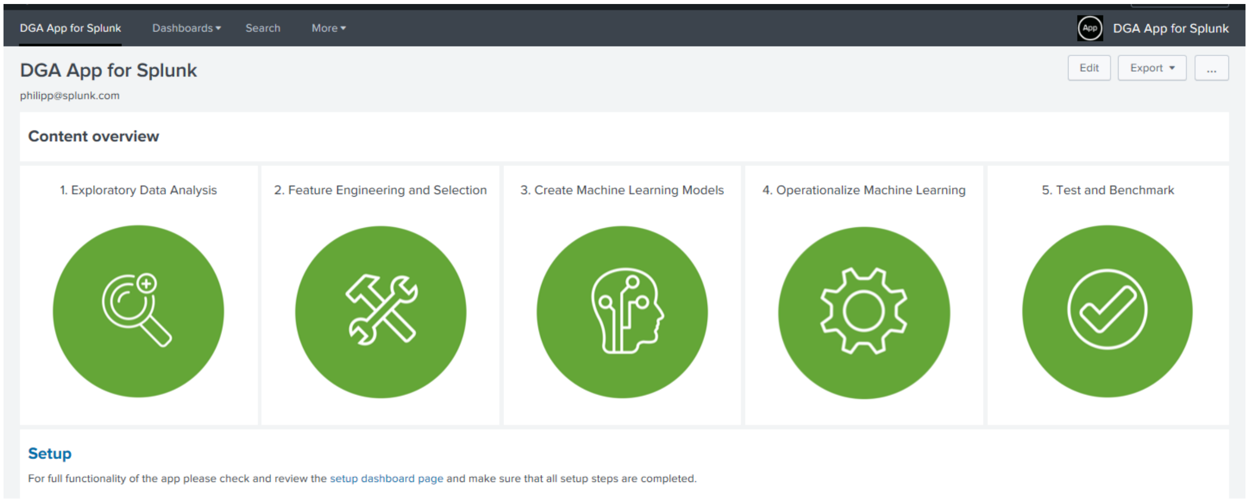
The Machine Learning Toolkit (MLTK) to build a DGA Application concepts *can be applied to create any type of Machine Learning Programs and Application*. Splunk’s DGA App provides *a clean user interface, dashboard, and training set* that *executes most of the backend calculations for simplicity*.

OVERVIEW

Follow the setup guideline under the dashboard for Splunk's DGA App to have full access.[5]

To begin: install app dependencies for full functionality

- Splunk Machine Learning Toolkit 2.4
- URL Toolbox App
- 3D Scatterplot
- Parallel coordinates



DGA App for Splunk

philipp@splunk.com

Content overview

1. Exploratory Data Analysis

2. Feature Engineering and Selection

3. Create Machine Learning Models

4. Operationalize Machine Learning

5. Test and Benchmark

Setup

For full functionality of the app please check and review the setup dashboard page and make sure that all setup steps are completed.

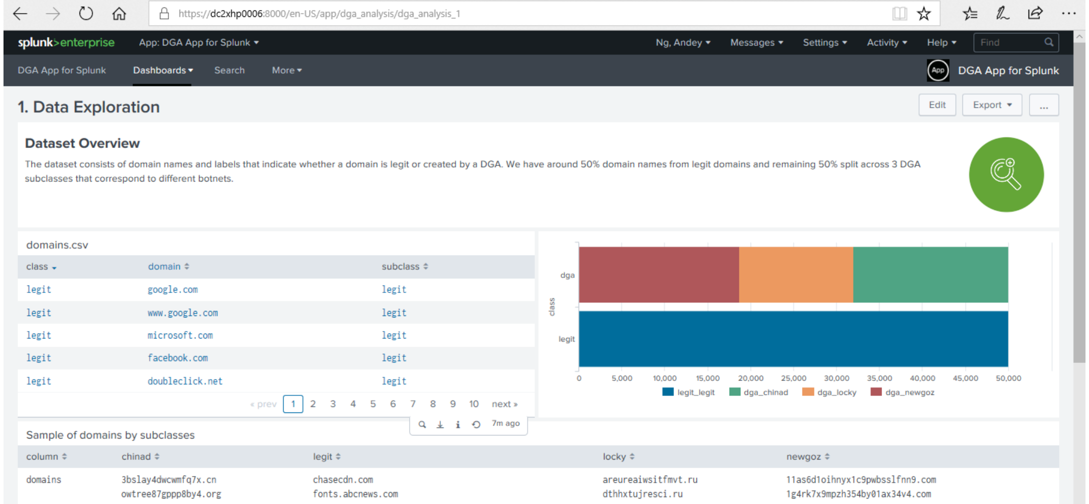
The DGA App MLTK consists of 5 Steps explained in this writeup:

1. Exploratory Data Analysis
2. Feature Engineering and Selection
3. Create Machine Learning Models
4. Operationalizing Machine Learning
5. Testing and Benchmark

[5] Splunk's DGA App Tutorial: https://www.youtube.com/watch?v=1ctPStvI3BY

1. Exploratory Data Analysis

This section creates mathematical models that help you visualize and analyze if there is a clear line or relationship in the data with the different colors in the graph.



1. Data Exploration

Dataset Overview

The dataset consists of domain names and labels that indicate whether a domain is legit or created by a DGA. We have around 50% domain names from legit domains and remaining 50% split across 3 DGA subclasses that correspond to different botnets.

domains.csv

class	domain	subclass
legit	google.com	legit
legit	www.google.com	legit
legit	microsoft.com	legit
legit	facebook.com	legit
legit	doubleclick.net	legit

Sample of domains by subclasses

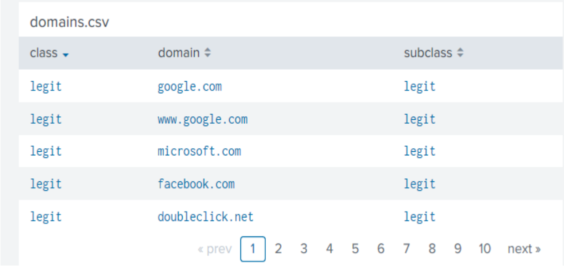
column	chinad	legit	locky	newgoz
domains	3bslay4dwcwmfq7x.cn owtree87gppp8by4.org	chasecdn.com fonts.abcnews.com	areureaiwsitfmvt.ru dthhxtujresci.ru	11as6d1oihnyx1c9pwbsslfnn9.com 1g4rk7x9mpzh354by01ax34v4.com

Dataset Overview

The first table on the left-hand side shown below is a list of 50,000 legitimate domains and 50,000 DGA domains that Splunk has identified and provided for us.

These provided domains serve as baseline data. You can add/tag more domains in the Testing and Benchmark Section.

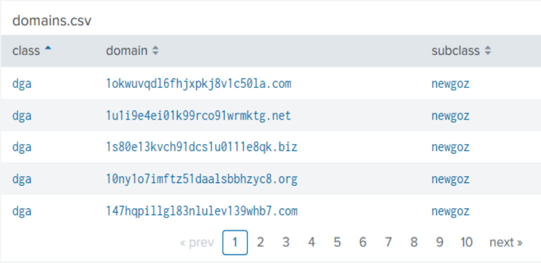
Legit Domain



domains.csv

class	domain	subclass
legit	google.com	legit
legit	www.google.com	legit
legit	microsoft.com	legit
legit	facebook.com	legit
legit	doubleclick.net	legit

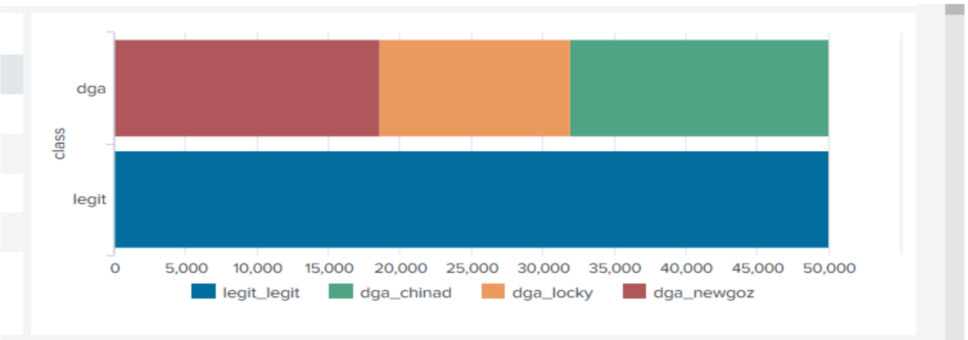
DGA Domain



domains.csv

class	domain	subclass
dga	1okwuvqdl6fhjxpkj8v1c50la.com	newgoz
dga	1u1i9e4ei01k99rco91wrmktg.net	newgoz
dga	1s80e13kvch91dcs1u0111e8qk.biz	newgoz
dga	10ny1o7imftz51daalsbbhzyc8.org	newgoz
dga	147hqpillgl83nlulev139whb7.com	newgoz

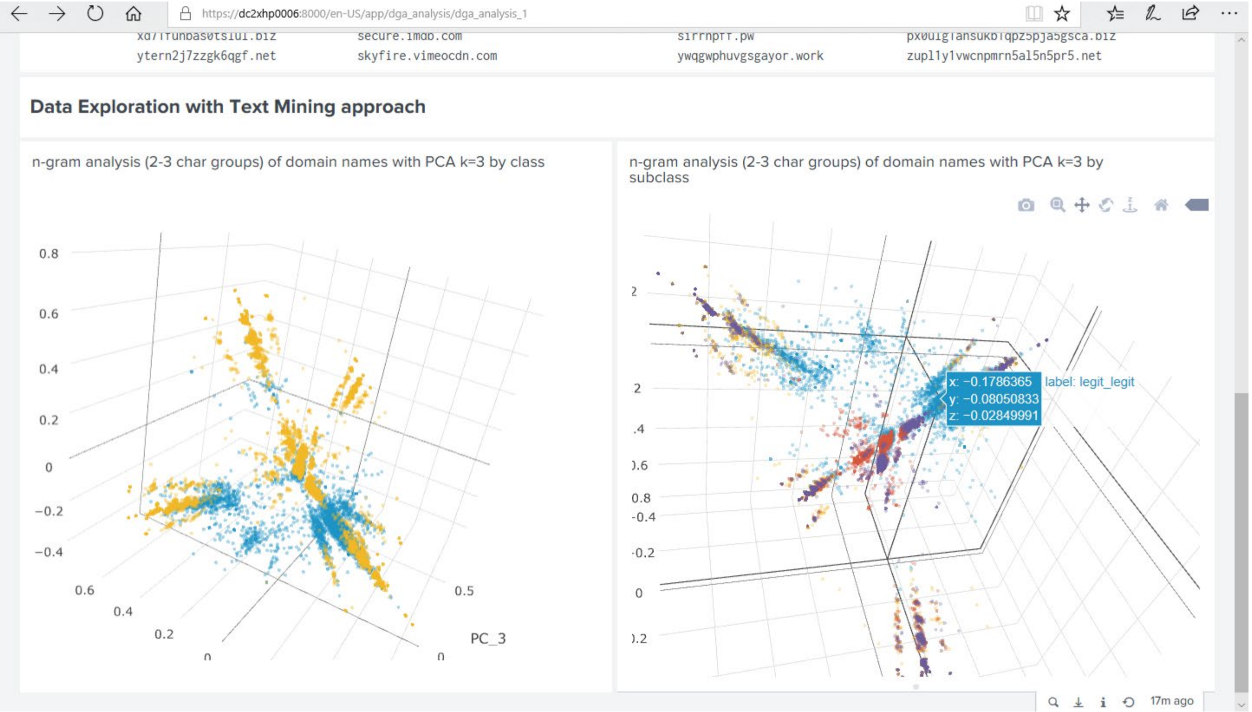
The graph below is a visual that can help you clearly see the distribution of classes and subclasses.



Data Exploration with Text Mining Approach

The second half of the page contains 3D scatterplots that takes the data from above to help you analyze and (possibly) draw conclusions about the relationships between the data based on the different color.

Identifying and cleaning this data helps us identify ***IF*** machine learning can be used here, and which type of machine learning algorithms that we can use. (Refer to Splunk's MLTK[6] documentation for more info).



https://dc2xhp0006:8000/en-US/app/dga_analysis/dga_analysis_1

```
xd7lfunbas0tslul.biz     secure.imdb.com          sirrnpff.pw              px0uiglansukblqpz5pja5gsca.biz
ytern2j7zzgk6qgf.net     skyfire.vimeocdn.com     ywqgwphuvgsgayor.work    zupl1y1vwcnpmrn5al5n5pr5.net
```

Data Exploration with Text Mining approach

n-gram analysis (2-3 char groups) of domain names with PCA k=3 by class

n-gram analysis (2-3 char groups) of domain names with PCA k=3 by subclass

The n-gram analysis below is a summary from Splunk's MLTK that explains what type of approach and algorithm it is using to create the 3D scatterplot.

For instance, one ML approach is the "Text Mining Approach" is an NLP[7] concept. This section uses a ***TFIDF*** *(Text Frequency Inverse Doc Frequency)*[8] approach that gives a word a score based ont the wieght of its frequency.

[6] Splunk's Machine Learning Toolkit (MLTK) : https://splunkbase.splunk.com/app/2890/

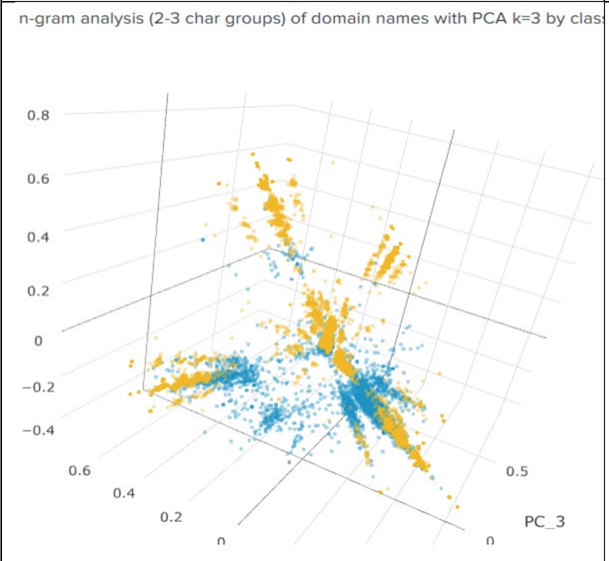
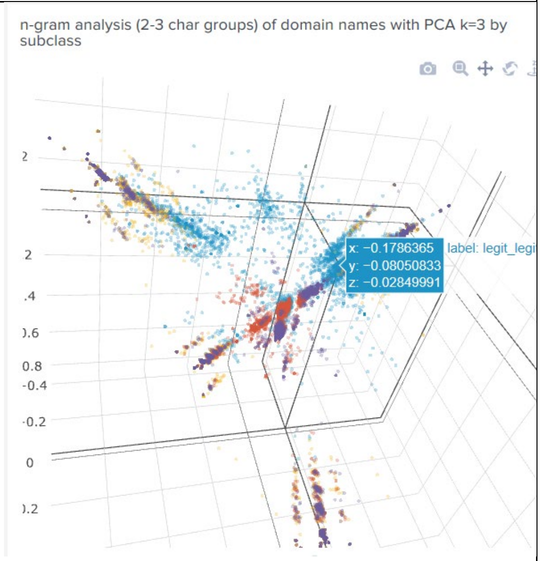
[7] Natural Language Processing (NLP): form of machine learning that understands our languages (English/Spanish etc.)

[8] TFIDF (Text Frequency Inverse Document Frequency): importance of words based on relevancy

This provided section refers to n-gram[9] modeling which creates a dataset of permutations of adjacent items (in this case characters) in the domain name. It takes a window size (2-3 characters), and creates a dataset that allows for you to identify patterns based on the frequency of these permutations.

Each dimension on the 3D graph is represented by the different PCA[10] (Principal Component Analysis) value.

Machine Learning is all about statistical probablity and prediction based on the previous patterns found, thus the n-gram model and the clustering of 3D graphs will help create distinctions between groups (DGA and legitimate domains).

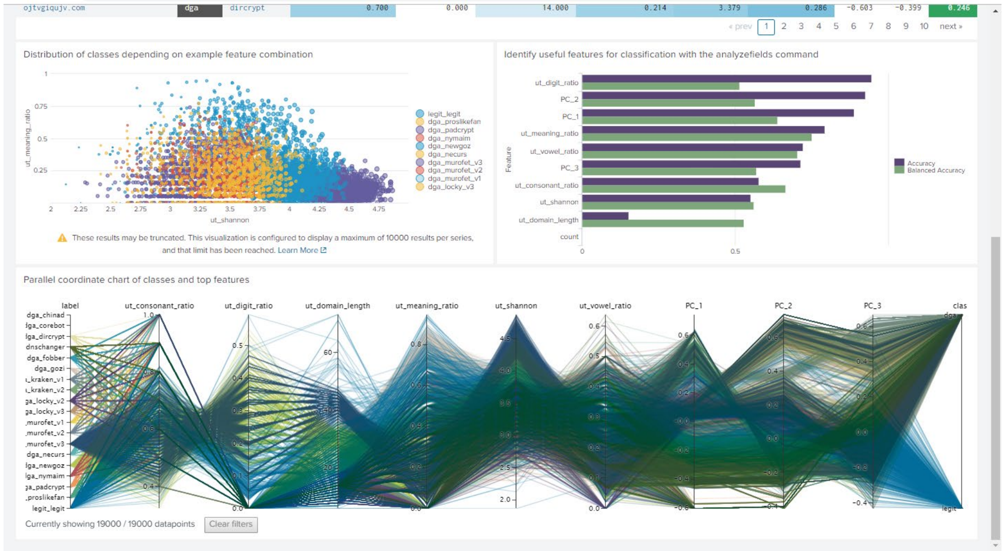
By Class	By Subclass
	
The class column identifies if the website is legit or DGA.	The subclass identifies where the attacks are coming from and the type of attack algorithm used (i.e. newgoz or chinad).

[9] N-gram model: A type of NLP approach that creates permutation of adjacent characters/words to help find patterns for ML

[10] Principal Component Analysis (PCA): a method of reducing large sets of variables to a more concise, informative descriptor without losing the original content

2. Feature Engineering and Selection

This section provides a sorted table of information of enriched data. The purpose of selection is to reduce the irrelevant information we're feeding into the machine learning model.



ojtvgiqujv.com	dga	dircrypt	0.700	0.000	14.000	0.214	3.379	0.286	-0.603	-0.399	0.246

« prev 1 2 3 4 5 6 7 8 9 10 next »

Distribution of classes depending on example feature combination

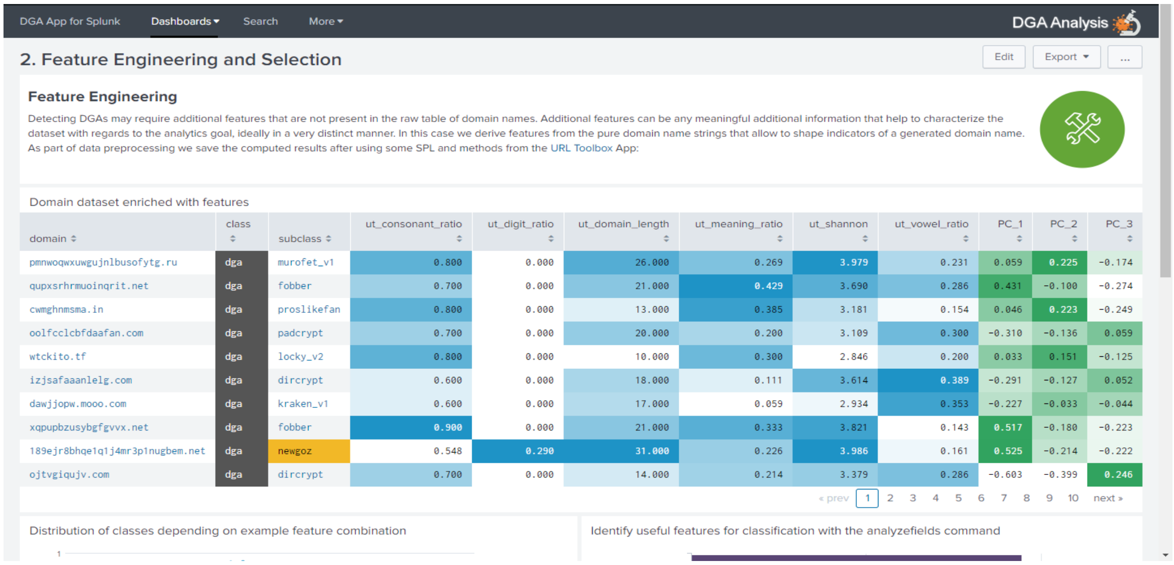
These results may be truncated. This visualization is configured to display a maximum of 10000 results per series, and that limit has been reached. Learn More

Identify useful features for classification with the analyzefields command

Parallel coordinate chart of classes and top features

Currently showing 19000 / 19000 datapoints Clear filters

In the previous section in the Exploratory Data Analysis Section, we received the PC1, PC2, and PC3 values that represent the 3 dimensions of the PCA values in the graphs above. The *URL Toolkit* enriches the string by providing the ratio of consonants, digits, meaning, vowels, and Shannon entropy index[11] of each domain.



2. Feature Engineering and Selection

Feature Engineering

Detecting DGAs may require additional features that are not present in the raw table of domain names. Additional features can be any meaningful additional information that help to characterize the dataset with regards to the analytics goal, ideally in a very distinct manner. In this case we derive features from the pure domain name strings that allow to shape indicators of a generated domain name. As part of data preprocessing we save the computed results after using some SPL and methods from the URL Toolbox App:

Domain dataset enriched with features

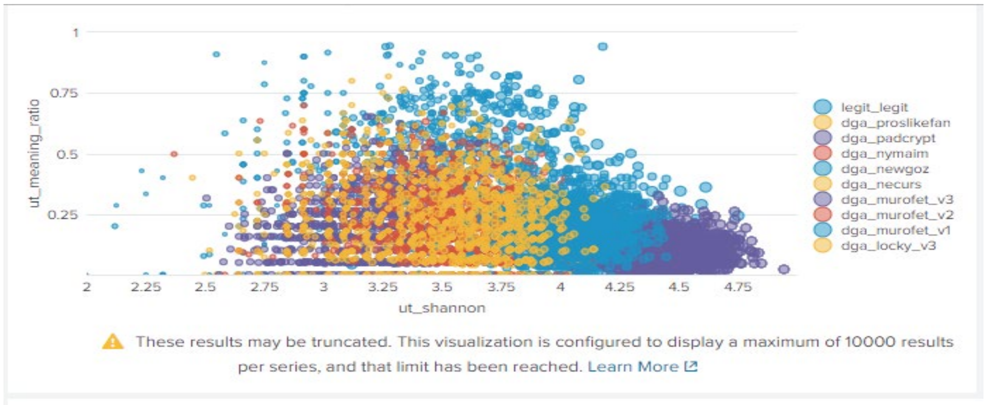
domain	class	subclass	ut_consonant_ratio	ut_digit_ratio	ut_domain_length	ut_meaning_ratio	ut_shannon	ut_vowel_ratio	PC_1	PC_2	PC_3
pmnwoqwxuwgujnlbusofytg.ru	dga	murofet_v1	0.800	0.000	26.000	0.269	3.979	0.231	0.059	0.225	-0.174
qupxsrhrmuoingrit.net	dga	fobber	0.700	0.000	21.000	0.429	3.690	0.286	0.431	-0.100	-0.274
cwmghnmsma.in	dga	proslikefan	0.800	0.000	13.000	0.385	3.181	0.154	0.046	0.223	-0.249
oolfcclcbfdaafan.com	dga	padcrypt	0.700	0.000	20.000	0.200	3.109	0.300	-0.310	-0.136	0.059
wtckito.tf	dga	locky_v2	0.800	0.000	10.000	0.300	2.846	0.200	0.033	0.151	-0.125
izjsafaaanlelg.com	dga	dircrypt	0.600	0.000	18.000	0.111	3.614	0.389	-0.291	-0.127	0.052
dawjjopw.mooo.com	dga	kraken_v1	0.600	0.000	17.000	0.059	2.934	0.353	-0.227	-0.033	-0.044
xqpupbzusybgfgvvx.net	dga	fobber	0.900	0.000	21.000	0.333	3.821	0.143	0.517	-0.180	-0.223
189ejr8bhqe1q1j4mr3p1nugbem.net	dga	newgoz	0.548	0.290	31.000	0.226	3.986	0.161	0.525	-0.214	-0.222
ojtvgiqujv.com	dga	dircrypt	0.700	0.000	14.000	0.214	3.379	0.286	-0.603	-0.399	0.246

« prev 1 2 3 4 5 6 7 8 9 10 next »

Distribution of classes depending on example feature combination

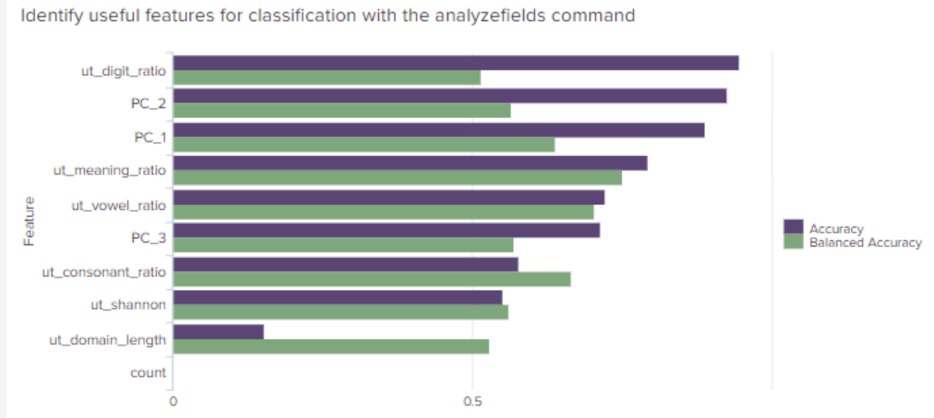
Identify useful features for classification with the analyzefields command

[11] Shannon Index: quantifying the entropy (uncertainty/information content) in a string



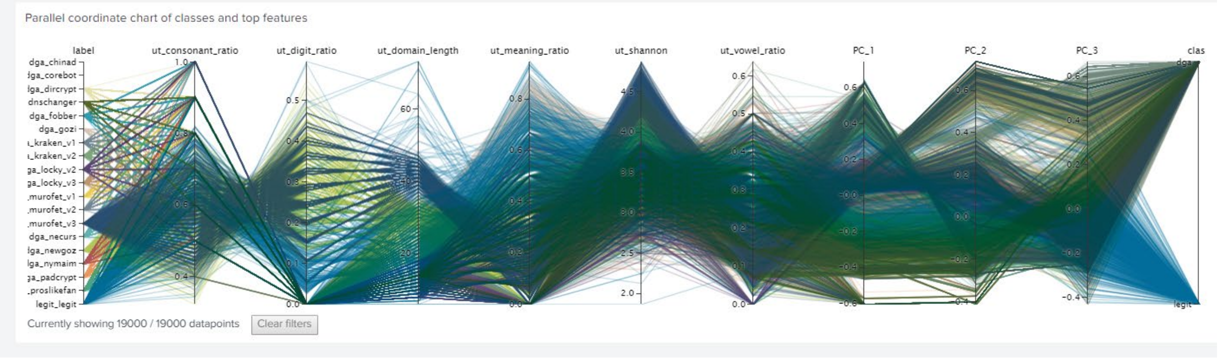
⚠ These results may be truncated. This visualization is configured to display a maximum of 10000 results per series, and that limit has been reached.

The chart takes 2 different dimensions (in this case it's Shannon Entropy vs Meaning Ratio) to see how the dimensions/variables separate to identify a relationship.



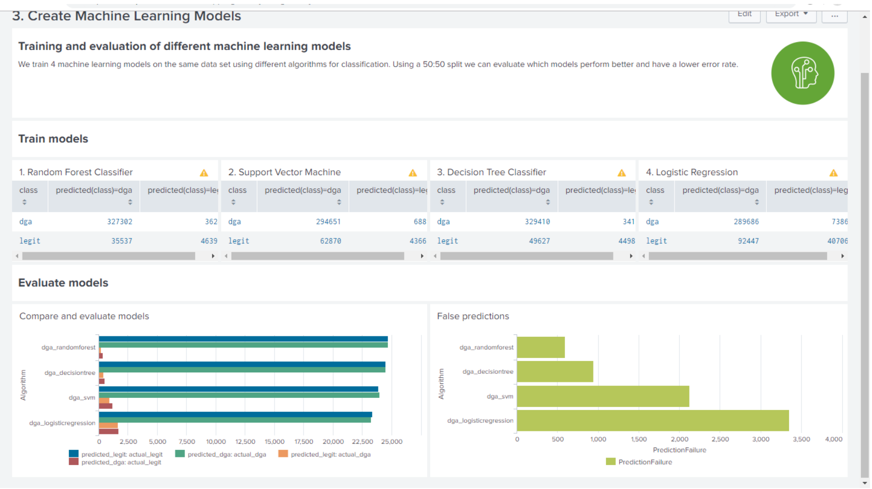
This chart contains a command called the *analyzefields* command which provides which the balanced accuracy and singular accuracy contribution to our dataset.

The Parallel Coordinate chart checks to see which features (variables) should be used. For example, some DGA domains will pass through the consonant ratio variable, so you would need other variables to refine the data.



3. Create Machine Learning Models

This section was intended to use different Machine Learning algorithms to determine the number of false positives (failed predictions).



3. Create Machine Learning Models

Training and evaluation of different machine learning models

We train 4 machine learning models on the same data set using different algorithms for classification. Using a 50:50 split we can evaluate which models perform better and have a lower error rate.

Train models

1. Random Forest Classifier ⚠

class	predicted(class)=dga	predicted(class)=le
dga	327302	362
legit	35537	4639

2. Support Vector Machine ⚠

class	predicted(class)=dga	predicted(class)=le
dga	294651	688
legit	62870	4366

3. Decision Tree Classifier ⚠

class	predicted(class)=dga	predicted(class)=le
dga	329410	341
legit	49627	4498

4. Logistic Regression ⚠

class	predicted(class)=dga	predicted(class)=leg
dga	289686	7386
legit	92447	40706

Evaluate models

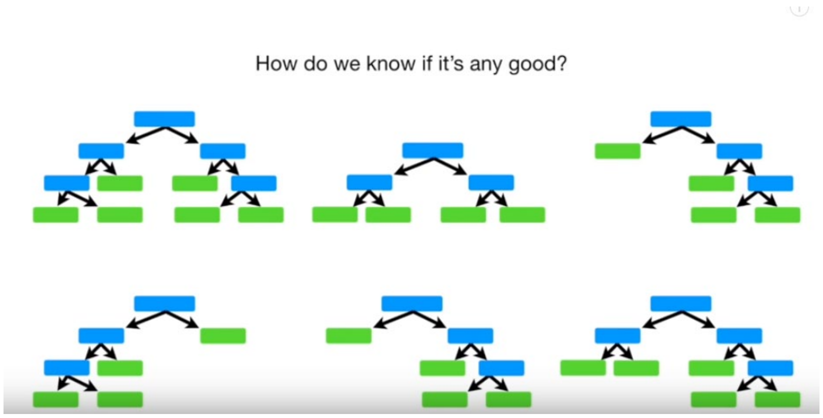
Compare and evaluate models

False predictions

There are many other types of Machine Learning Models, with algorithms from Sci-kit Learn[12], Splunk's MLTK, and other API's. This section provides 4 of the most widely-used machine learning algorithms:

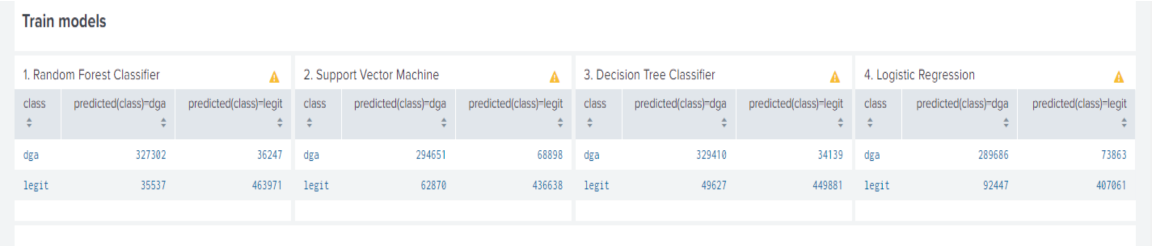
1. *Random Forest Classifier*
2. *Support Vector Machine (SVM)*
3. *Decision Tree Classifier*
4. *Logistic Regression*

Below is a chart that describes how many decision trees in machine learning algorithms work.



[12] Sci-kit Learn: one of the most commonly used machine learning libraries for Python

Each table contains a training set with the number of correctly/incorrectly predicted domain types to help inform you on which algorithm you might want to use.



Train models

1. Random Forest Classifier ⚠

class	predicted(class)=dga	predicted(class)=legit
dga	327302	36247
legit	35537	463971

2. Support Vector Machine ⚠

class	predicted(class)=dga	predicted(class)=legit
dga	294651	68898
legit	62870	436638

3. Decision Tree Classifier ⚠

class	predicted(class)=dga	predicted(class)=legit
dga	329410	34139
legit	49627	449881

4. Logistic Regression ⚠

class	predicted(class)=dga	predicted(class)=legit
dga	289686	73863
legit	92447	407061

Logistic Regression[13] use cost function analysis to classify and categorize variables. This type of machine learning utilizes the gradient descent concept that uses both independent and dependent variables. Though this is a well-supported machine learning model, it is not ideal for the DGA predicitve model, as DGAs do not rely on depedent variables.

Support Vector Machines (SVM)[14] is datapoint classification on a hyperplane most commonly used when less computation power is required. However, SVMs not the most ideal for this project because SVM best explains the marginal difference between 2 variables, and this DGA project simply requires a "yes/no" classification.

Decision Trees[15] cover possible consequences, including chance event outcomes, resource costs, and utility. The DGA project is identifying the chances and likelihood of a domain being legitimate or fake; however, decision trees are susceptible to biases and overfitting.

Thus, the Random Forest Classifiers are most optimal machine learning model for this project, as it uses a multitude of decision trees and creates a verdict based on the consensus of those trees, reducing the model to contain the least amount of biases.[16]

[13] More details about Logistic Regression

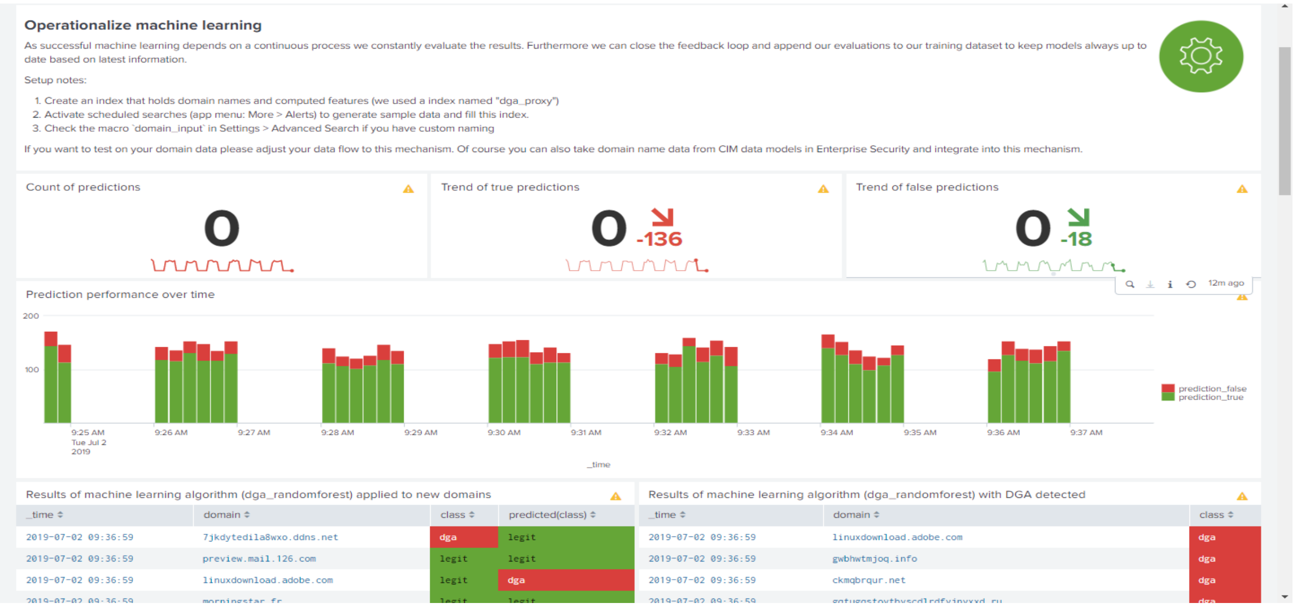
[14] More details about Support Vector Machines

[15] More details about Decision Tree Classifiers

[16] More detail about Random Forests

4. Operationalize Machine Learning

This section allows you to test each machine learning model with real-time data to evaluate accurate and false positive predictions.



Operationalize machine learning

As successful machine learning depends on a continuous process we constantly evaluate the results. Furthermore we can close the feedback loop and append our evaluations to our training dataset to keep models always up to date based on latest information.

Setup notes:

1. Create an index that holds domain names and computed features (we used a index named "dga_proxy")
2. Activate scheduled searches (app menu: More > Alerts) to generate sample data and fill this index.
3. Check the macro `domain_input` in Settings > Advanced Search if you have custom naming

If you want to test on your domain data please adjust your data flow to this mechanism. Of course you can also take domain name data from CIM data models in Enterprise Security and integrate into this mechanism.

Results of machine learning algorithm (dga_randomforest) applied to new domains

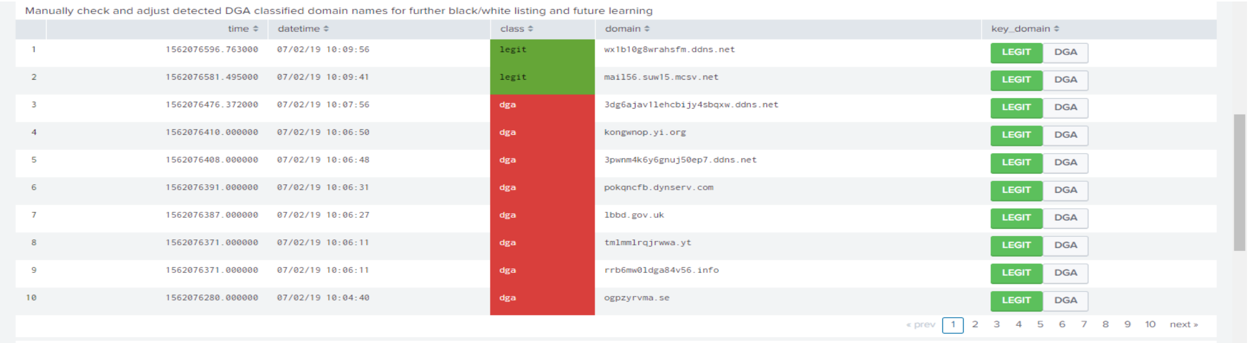
_time ⇕	domain ⇕	class ⇕	predicted(class) ⇕
2019-07-02 09:36:59	7jkdytedila8wxo.ddns.net	dga	legit
2019-07-02 09:36:59	preview.mail.126.com	legit	legit
2019-07-02 09:36:59	linuxdownload.adobe.com	legit	dga
2019-07-02 09:36:59	morningstar.fr	legit	legit

Results of machine learning algorithm (dga_randomforest) with DGA detected

_time ⇕	domain ⇕	class ⇕
2019-07-02 09:36:59	linuxdownload.adobe.com	dga
2019-07-02 09:36:59	gwbhwtmjoq.info	dga
2019-07-02 09:36:59	ckmqbrqur.net	dga
2019-07-02 09:36:59	gotugstovthyscdlrdfvinvxxd.ru	dga

The table below allows security analysts to *manually correct* the machine learning predictions to update and teach the training set (data model). This refinement will optimize the algorithm's accuracy for future classification predictions.

For instance, if the domain is falsely classified as a legitamate domain, the analyst can click the "DGA button" on the right to retrain the algorithm's data.



Manually check and adjust detected DGA classified domain names for further black/white listing and future learning

	time ⇕	datetime ⇕	class ⇕	domain ⇕	key_domain ⇕
1	1562076596.763000	07/02/19 10:09:56	legit	wx1b10g8wrahsfm.ddns.net	LEGIT DGA
2	1562076581.495000	07/02/19 10:09:41	legit	mail56.suw15.mcsv.net	LEGIT DGA
3	1562076476.372000	07/02/19 10:07:56	dga	3dg6ajav1lehcbijy4sbqxw.ddns.net	LEGIT DGA
4	1562076410.000000	07/02/19 10:06:50	dga	kongwnop.yi.org	LEGIT DGA
5	1562076408.000000	07/02/19 10:06:48	dga	3pwnm4k6y6gnuj50ep7.ddns.net	LEGIT DGA
6	1562076391.000000	07/02/19 10:06:31	dga	pokqncfb.dynserv.com	LEGIT DGA
7	1562076387.000000	07/02/19 10:06:27	dga	lbbd.gov.uk	LEGIT DGA
8	1562076371.000000	07/02/19 10:06:11	dga	tmlmmlrqjrwwa.yt	LEGIT DGA
9	1562076371.000000	07/02/19 10:06:11	dga	rrb6mw0ldga84v56.info	LEGIT DGA
10	1562076280.000000	07/02/19 10:04:40	dga	ogpzyrvma.se	LEGIT DGA

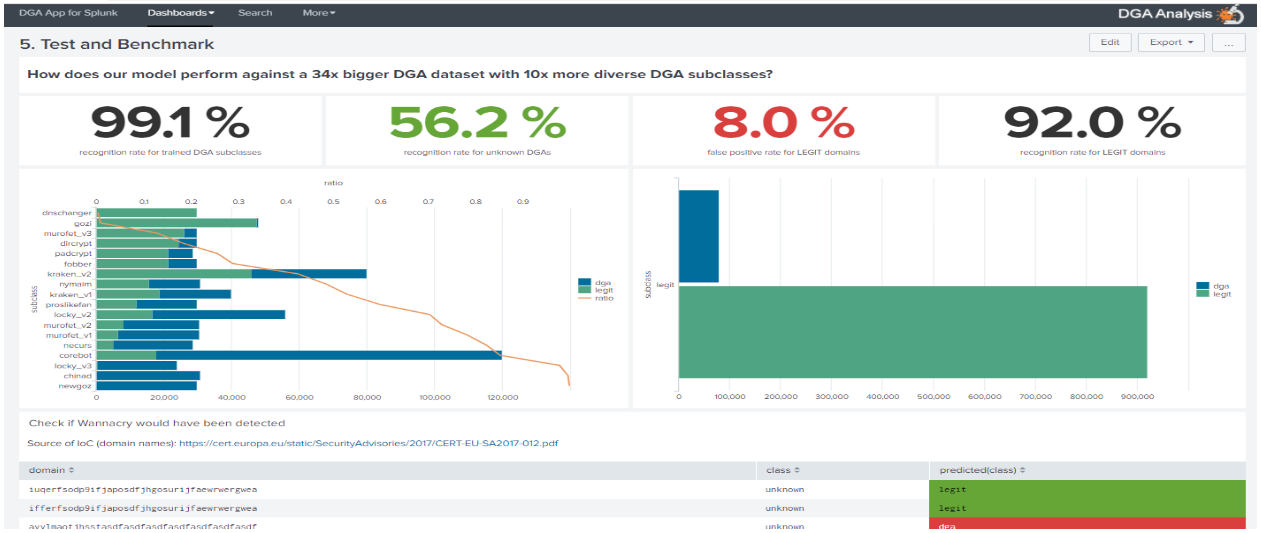
After this refinement, the algorithm will whitelist the legitamate domains and blacklist the DGA domains, adding it to the new model.[17]

[17] Refer to the first figure on the first page for a visual understanding of the DGA's machine learning refinement

5. Test and Benchmark

Working with new larger datasets will assure that the DGA App is hardcoded (using brute force combinations), and that the DGA App is truly a flexible machine learning model.

Based on the charts below, you can analyze which features need more refinement to retrain the model with larger datasets. This section is meant to identify the accuracy of the DGA algorithm on a larger dataset.



5. Test and Benchmark

How does our model perform against a 34x bigger DGA dataset with 10x more diverse DGA subclasses?

99.1 %	56.2 %	8.0 %	92.0 %
recognition rate for trained DGA subclasses	recognition rate for unknown DGAs	false positive rate for LEGIT domains	recognition rate for LEGIT domains

Check if Wannacry would have been detected

Source of IoC (domain names): https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf

domain ⇕	class ⇕	predicted(class) ⇕
iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea	unknown	legit
ifferfsodp9ifjaposdfjhgosurijfaewrwergwea	unknown	legit
ayylmaotjhsstasdfasdfasdfasdfasdfasdfasdf	unknown	dga

Future Outlook & Reflection

There are some feasibility drawbacks with Splunk and the machine learning model. Currently, Splunk does not support Keras or Tensorflow, two of the leading machine learning libraries. Additionally, bots can quickly and easily obfuscate new domains faster than most machine learning models can react.

A group of PhD students at Georgia Tech proposed a detection system, Pleiades which sits between the network machines and the recursive DNS server and analyzes DNS queries for domain names that result in Name Error responses to identify if the IP address exists.[18] However, that is still being tested and is difficult to implement at an industry and corporate setting.

[18] More details about Pleiades Detection System